Polynomial weights and code constructions

Massey, J; Costello, D; Justesen, Jørn

Published in:
I E E E Transactions on Information Theory

Publication date:
1973

Document Version
Publisher's PDF, also known as Version of record

With $n_0 = 4$, $\mu = 10$, n can be increased to $16 \cdot 40 = 640$ bits, the number of redundant symbols $r = 160$, $k = 640 - 160 = 480$ and

$$P_{\oplus} = \frac{\sum_{i=1}^{3} \binom{16}{i}}{2^{40}} \cong 7 \cdot 10^{-10}.$$

To correct all phased burst errors with $\Delta = 3$ using the Reed–Solomon code over $GF(2^{30})$ we would need $r = 7 \cdot 30 = 210$ redundant bits, but the block length in this case can be much longer. Computer implementation of operations over $GF(2^3)$ is substantially simpler than over $GF(2^{30})$.

Acknowledgment

The author expresses his appreciation to Prof. J. K. Wolf for very helpful discussions and criticism. The author is also indebted to the referees for their valuable comments and suggestions.

References

[1] W. W. Peterson, *Error-Correcting Codes*. New York: M.I.T. Press and Wiley, 1961.
[2] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed. Cambridge, Mass.: M.I.T. Press, 1972.
[3] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.
[4] V. D. Kolesnik and E. T. Mironchikov, *Decoding of Cyclic Codes*. (in Russian). Moscow: Svyaz, 1968.
[5] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, Mass.: M.I.T. Press, 1963.
[6] J. L. Massey, *Threshold Decoding*. Cambridge, Mass.: M.I.T. Press, 1963.
[7] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, Mass.: M.I.T. Press, 1966.
[8] S. I. Samoylenko, *Error-Correcting Codes* (in Russian). Moscow: Nauka, 1966.
[9] J. K. Wolf, "Adding two information symbols to certain nonbinary BCH codes and some applications," *Bell Syst. Tech. J.*, vol. 48, Sept. 1969.
[10] S. Lin, *An Introduction To Error-Correcting Codes*. Englewood Cliffs, N.J.: Prentice-Hall, 1970.

Polynomial Weights and Code Constructions

JAMES L. MASSEY, DANIEL J. COSTELLO, JR., AND JØRN JUSTESEN

***Abstract*—For any nonzero element c of a general finite field $GF(q)$, it is shown that the polynomials $(x - c)^i$, $i = 0,1,2,\cdots$, have the "weight-retaining" property that any linear combination of these polynomials with coefficients in $GF(q)$ has Hamming weight at least as great as that of the minimum degree polynomial included. This fundamental property is then used as the key to a variety of code constructions including 1) a simplified derivation of the binary Reed–Muller codes and, for any prime p greater than 2, a new extensive class of p-ary "Reed–Muller codes," 2) a new class of "repeated-root" cyclic codes that are subcodes of the binary Reed–Muller codes and can be very simply instrumented, 3) a new class of constacyclic codes that are subcodes of the p-ary "Reed–Muller codes," 4) two new classes of binary convolutional codes with large "free distance" derived from known binary cyclic codes, 5) two new classes of long constraint length binary convolutional codes derived from 2^r-ary Reed–Solomon codes, and 6) a new class of q-ary "repeated-root" constacyclic codes with an algebraic decoding algorithm.**

Manuscript received January 27, 1972. This work was supported in part by NASA Grant NGL 15-004-026 and by National Science Foundation Grant GK-5265.

J. L. Massey was on leave at the Laboratory for Communication Theory, Technical University of Denmark, Lyngby, Denmark. He is now with the University of Notre Dame, Notre Dame, Ind.
D. J. Costello, Jr., is with the Department of Electrical Engineering, Illinois Institute of Technology, Chicago, Ill.
J. Justesen is with the Laboratory for Communication Theory, Technical University of Denmark, Lyngby, Denmark.

I. Introduction

IN THIS paper, it is shown that the polynomials $(x - c)^i$, $i = 0,1,2,\cdots$, where c is any nonzero element of $GF(q)$, have the fundamental property, which we term "weight-retaining," that any linear combination of these polynomials with coefficients in $GF(q)$ has Hamming weight at least as great as that of the minimum degree polynomial included. This is proved separately in Section II-A for the case where $GF(q)$ has characteristic $p = 2$ since the binary case has a simplicity lacking in general and since it has the most interesting applications. The applications to 1) Reed–Muller codes, 2) a new class of "repeated-root" binary cyclic codes, 3) two new classes of binary convolutional codes derived from binary cyclic codes, and 4) two new classes of binary convolutional codes derived from Reed–Solomon codes are given in Sections II-B, II-C, II-D, and II-E, respectively.

In Section III-A, we give a new class of "constacyclic" codes over fields $GF(q)$ with characteristic p greater than 2 that are "maximum distance separable" and have a simple algebraic decoding algorithm. This class of codes is then employed in Section III-B as the basis for an inductive proof

of the weight-retaining property for a general finite field $GF(q)$. Finally, in Section III-C, we give a new p-ary generalization of the Reed–Muller codes and a new class of constacyclic subcodes of these p-ary codes having the same minimum distance as the parent codes.

II. The Binary Case

A. The Weight-Retaining Property in Fields of Characteristic Two

Let c be a nonzero element of the finite field $GF(q)$ where $q = 2^r$ for some integer r. Since the polynomials $(x + c)^i$, $i = 0,1,2,\cdots$, include exactly one polynomial of each degree, they are a basis for the vector space of all polynomials over $GF(2^r)$ and hence every polynomial $P(x)$ over $GF(2^r)$ can be expressed uniquely as a linear combination of these polynomials. Hereafter, let $W[P(x)]$ denote the *Hamming weight* of $P(x)$, i.e., the number of its nonzero coefficients. The following theorem relates $W[P(x)]$ to the expansion of $P(x)$ in the above basis.

Theorem 1.1: Let I be any finite nonempty set of nonnegative integers with least integer $i_{\min}$ and let

$$P(x) = \sum_{i \in I} b_i(x + c)^i$$

where c and each b_i is a nonzero element of $GF(2^r)$. Then

$$W[P(x)] \geq W[(x + c)^{i_{\min}}]. \tag{1}$$

Proof: We proceed by induction on the greatest integer $i_{\max}$ in I. A simple check shows that (1) holds for $i_{\max} < 2^2$. We suppose then that (1) holds for $i_{\max} < 2^n$ and show that (1) holds for $i_{\max} < 2^{n+1}$.

Partition I into the sets I_0 and I_1, where I_0 contains those and only those i in I such that $i < 2^n$. Then

$$\sum_{i \in I_1} b_i(x + c)^i = (x + c)^{2^n} \sum_{i \in I_1} b_i(x + c)^{i - 2^n},$$

which, upon writing $P_1(x)$ for the summation on the right-hand side which is a polynomial of degree less than 2^n, becomes

$$\sum_{i \in I_1} b_i(x + c)^i = x^{2^n}P_1(x) + c^{2^n}P_1(x). \tag{2}$$

Similarly, write $P_0(x)$ as the polynomial of degree less than 2^n given by

$$P_0(x) = \sum_{i \in I_0} b_i(x + c)^i.$$

From (2) and the definitions of I_0 and I_1 we then have

$$P(x) = [P_0(x) + c^{2^n}P_1(x)] + x^{2^n}P_1(x). \tag{3}$$

Suppose first that $P_0(x) = 0$. Then, from (3), we have $W[P(x)] = 2W[P_1(x)]$. Since $P_1(x)$ has degree less than 2^n, we have from the induction hypothesis

$$W[P_1(x)] \geq W[(x + c)^{i_{\min} - 2^n}]. \tag{4}$$

But also

$$\begin{aligned} W[(x + c)^{i_{\min}}] &= W[(x + c)^{2^n}(x + c)^{i_{\min} - 2^n}] \\ &= W[(x^{2^n} + c^{2^n})(x + c)^{i_{\min} - 2^n}] \\ &= 2W[(x + c)^{i_{\min} - 2^n}] \end{aligned}$$

so that with the aid of (4) we have $W[P(x)] = 2W[P_1(x)] \geq W[(x + c)^{i_{\min}}]$, as was to be shown. Conversely, suppose that $P_0(x) \neq 0$. We then have from (3)

$$W[P(x)] \geq W[P_0(x)] \tag{5}$$

since any nonzero terms in $P_0(x)$ cancelled by the addition of $c^{2^n}P_1(x)$ must reappear as nonzero terms in $x^{2^n}P_1(x)$. Since $P_0(x)$ has degree less than 2^n, the induction hypothesis gives $W[P_0(x)] \geq W[(x + c)^{i_{\min}}]$, which, together with (5), yields (1), and the theorem is proved.

We remark that, for the special case $r = 1$, Theorem 1.1 is equivalent to showing that the binary $2^n \times 2^n$ matrix whose $(i + 1)$th row is the sequence of coefficients in $(x + 1)^i$, i.e., the $(i + 1)$th row of Pascal's triangle reduced modulo 2, has the property that any sum of its rows has Hamming weight at least as great as the uppermost row included in the sum. For $n = 3$, this matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

These matrices are of some importance in switching theory where Preparata [1] has pointed out other interesting properties of these matrices, including the fact that they are self-inverse.

B. Binary Reed–Muller Codes

We shall make frequent use of the following fact.

Lemma 1: Let c be a nonzero element of a finite field $GF(q)$ with characteristic p, and let i be a nonnegative integer with radix-p form $[i_{m-1},\cdots,i_1,i_0]$. Then

$$W[(x + c)^i] = \prod_{j=0}^{m-1} (i_j + 1) \tag{6a}$$

or, for the particular case where $p = 2$,

$$W[(x + 1)^i] = 2^{w(i)} \tag{6b}$$

where $w(i)$ is the number of 1's in the radix-2 form of i.

To prove this lemma, we first note that $W[(x + c)^i]$ is just the number of integers k, $0 \leq k \leq i$, such that the binomial coefficient $\binom{i}{k}$ is nonzero modulo p. But, by a theorem of Lucas [2, p. 113],

$$\binom{i}{k} \equiv \prod_{j=0}^{m-1} \binom{i_j}{k_j} \bmod p \tag{7}$$

where the i_j and the k_j are the digits in the radix-p forms of i and k, respectively and where, by convention, a binomial coefficient whose lower member exceeds its upper is zero. It then follows from (7) that there are exactly $i_j + 1$ choices for k_j, namely $0,1,2,\cdots,i_j$, such that the corresponding binomial coefficient in (7) is nonzero modulo p. Thus (6a) follows and the lemma is proved.

We are now in position to use Theorem 1.1 for a simple derivation of the binary Reed–Muller codes. Let m and u be any two positive integers such that $u \leq m$. Consider the binary matrix G with $n = 2^m$ columns whose rows contain the sequence of coefficients in $(x + 1)^i$ for all i such that $i < n$ and $w(i) \geq u$. The number of such i, i.e., the number of rows of G, is just

$$k = \sum_{j=u}^{m} \binom{m}{j} = \sum_{j=0}^{m-u} \binom{m}{j}.$$

For example, with $m = 3$ and $u = 2$, we have

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

We then take G to be the generator matrix of an (n,k) binary parity-check code. It follows from Theorem 1.1 that every nonzero codeword, i.e., every sum of one or more rows of G, has Hamming weight at least 2^u, since our choice of G ensures that such a sum corresponds to a sum of polynomials $(x + 1)^i$ for which $w(i_{\min}) \geq u$ and hence, by Lemma 1, $W[(x + 1)^{i_{\min}}] \geq 2^u$. Moreover, some rows of G have Hamming weight exactly 2^u so the minimum distance of the code is $d = 2^u$. This code is precisely the uth order Reed–Muller code of length $n = 2^m$ and, in fact, the rows that we have chosen for G are the same as those chosen by Reed [3] (except for a trivial reversal of each row).

The evaluation of k and d for the binary Reed–Muller codes as given here is a substantial simplification of past arguments.

C. A New Class of Binary Repeated-Root Cyclic Codes

For $2 \leq u \leq m$, consider the binary polynomial

$$g(x) = (x + 1)^i \tag{8}$$

where

$$i = 2^m - 2^{m-u+1} + 1. \tag{9}$$

The radix-2 form of i is then

$$[i_{m-1}, \cdots, i_1, i_0] = [1 \quad 1 \quad \cdots \quad 1 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 1] \tag{10}$$

where the run of 0's has length $m - u$. We note that $2^{m-1} < i < 2^m$, which implies that $g(x)$ divides $x^n + 1$ for $n = 2^m$ (since then $x^n + 1 = (x + 1)^n$) but for no smaller n. Hence, $g(x)$ generates a binary (n,k) cyclic code with $n = 2^m$ and $n - k = i$, which, from (9), gives $k = 2^{m-u+1} - 1$. The rows of the generator matrix $\hat{G}$ for this cyclic code may be chosen as $g(x)(x + 1)^j$ for for $j = 0,1,\cdots,k - 1$, or equivalently as $(x + 1)^j$ for $i \leq j < n$. But it follows from (10) that $w(j) \geq w(i)$ for $i \leq j < n = 2^m$ since the radix-2 form of j must have $u - 1$ leading 1's and at least one 1 in its last $m - u + 1$ positions. Thus the rows of $\hat{G}$ are a subset of the rows of G as given in Section 11-B. Hence the cyclic code generated by $g(x)$ is a subcode of the uth order binary Reed–Muller code having the same minimum distance as the parent code since $W[g(x)] = 2^u = d$. [Hereafter we call a cyclic code in which the irreducible factors and hence the roots of $g(x)$ have multiplicity greater than one a *repeated-root* cyclic code.] We have then proved the following.

Theorem 2: For $2 \leq u \leq m$, the $(n = 2^m, k = 2^{m-u+1} - 1)$ binary repeated-root cyclic code generated by $g(x) = (x + 1)^{n+k}$ is a subcode of the uth order binary Reed–Muller code having the same minimum distance $d = 2^u$ as the parent code.

Although the repeated-root cyclic codes of Theorem 2 are generally inferior to comparable Bose–Chaudhuri–Hocquenghem (BCH) codes, they have two interesting properties that might recommend their use in certain practical applications.

First, very simple syndrome-forming and decoding circuitry is possible for the repeated-root codes. The circuitry utilizes logical elements corresponding to the factor $(x + 1)$ in combinations that lend themselves to implementation with integrated circuits. Consider the circuit of Fig. 1(a). It may be readily checked that if a polynomial $P(x) = P_{n-1}x^{n-1} + \cdots + P_1x + P_0$ is read into this circuit with higher degree coefficients leading, then the contents

$$s(x) = s_0 + s_1x + \cdots + s_{n-k-1}x^{n-k-1}$$

when P_j is the only nonzero coefficient will be $P_j(x + 1)^j \cdot \mod (x^{n-k})$; hence, by linearity, the response for general $P(x)$ will be

$$s(x) = P(x + 1) \bmod (x^{n-k})$$

[where here and hereafter we write $P(x) \bmod Q(x)$ for the remainder when the polynomial $P(x)$ is divided by the polynomial $Q(x)$.] In particular, when $P(x) = f(x) + e(x)$ where $f(x) = a(x)g(x) = a(x)(x + 1)^{n-k}$ is a codeword in the repeated-root code and $e(x) = e_0 + e_1x + \cdots + e_{n-1}x^{n-1}$ is the channel error-pattern, we have

$$\begin{aligned} s(x) &= e(x + 1) \bmod (x^{n-k}) \\ &= \sum_{j=0}^{n-1} e_j(x + 1)^j \bmod (x^{n-k}), \end{aligned} \tag{11}$$

which shows that $s(x)$ depends only on the error pattern and is thus a true syndrome. Suppose then that one has realized a logical function F that forms the decoding estimate $\hat{e}_{n-1}$ of the leading error digit e_{n-1} from the syndrome $s(x)$. After further i shifts of the logic in Fig. 1(a), the contents of the syndrome register become

$$\text{“}s(x)\text{”} = \sum_{j=0}^{n-1} e_{j-i}(x + 1)^j \bmod (x^{n-k})$$

[where we understand $e_{-h} = e_{n-h}$] so that the same function F will then be forming the corresponding estimate $\hat{e}_{n-i-1}$ of e_{n-i-1}. Thus, reminiscent of the technique first proposed by Meggitt [4] for cyclic codes, a complete decoder for the repeated-root code can be implemented as shown in Fig. 1(b). The connection shown dotted in this decoder is included if it is desired to “remove” the effect of correctly decoded error digits from the syndrome so that the syndrome

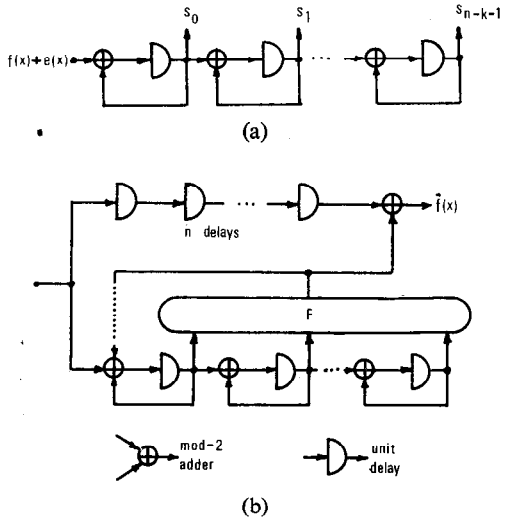

Fig. 1. (a) Syndrome-forming circuit for the binary repeated-root cyclic codes of Theorem 2. (b) Complete decoder.

contents must be all 0's after successful decoding of the complete block.

Second, since the repeated-root codes are subcodes of the Reed–Muller codes having the same minimum distance as the parent codes, Reed's majority logic decoding algorithm [3] can be used for a simple realization of the decoding function F in Fig. 1(b). For instance, for the (8,3) repeated-root code with $d = 2^2 = 4$, it may be readily checked that s_3, s_4 and $s_0 + s_3 + s_4$ form a set of three parity checks orthogonal on $e_{n-1} = e_7$ [5] so that F may be realized to conform with the decoding rule: $\hat{e} = 0$ if none or one of these checks have value 1, $\hat{e} = 1$ if all three have value 1, and detection of 2 or more errors announced if two of these checks have value 1.

D. Construction of Binary Convolutional Codes from Binary Cyclic Codes

As we now proceed to show, Theorem 1.1 provides the key for using known cyclic codes to construct convolutional codes with large "free distance." To facilitate this discussion, we first recast Theorem 1.1 into the following two equivalent forms.

Theorem 1.2: For any polynomial $Q(x)$ over $GF(2^r)$, any nonzero element c of $GF(2^r)$, and any nonnegative integer N,

$$W[Q(x)(x + c)^N] \geq W[(x + c)^N] \cdot W[Q(c)]. \quad (12)$$

Proof: We first expand $Q(x)$ as

$$Q(x) = \sum_{i=0}^{t} b_i(x + c)^i$$

and note that $Q(c) = b_0$. If $b_0 = 0$, then $W[Q(c)] = 0$ and (12) holds trivially. If $b_0 \neq 0$, then $W[Q(c)] = 1$ and taking $P(x)$ in Theorem 1.1 to be $(x + c)^N Q(x)$, we have $i_{\min} = N$ so that again (12) holds.

Theorem 1.3: For any polynomial $P(x)$ over $GF(2^r)$, any nonzero element c of $GF(2^r)$, and any nonnegative integers n and N,

$$W[P(x)(x^n + c)^N] \geq W[(x + c)^N] \cdot W[P(x) \bmod (x^n + c)]. \quad (13)$$

Proof: Letting $P(x) = Q_1(x^n) + xQ_2(x^n) + \cdots + x^{n-1}Q_n(x^n)$, we have

$$W[P(x)(x^n + c)^N] = \sum_{i=1}^{n} W[Q_i(x^n)(x^n + c)^N]. \quad (14)$$

Now, identifying x^n on the right-hand side of (14) with x in Theorem 1.2, we obtain

$$\begin{aligned} W[P(x)(x^n + c)^N] &\geq \sum_{i=1}^{n} W[Q_i(c)] \cdot W[(x + c)^N] \\ &= W[(x + c)^N] \cdot \sum_{i=1}^{n} W[Q_i(c)] \\ &= W[(x + c)^N] \cdot W\left[\sum_{i=1}^{n} x^{i-1}Q_i(c)\right] \end{aligned}$$

and (13) now follows upon noting that the last summation is just the polynomial $P(x) \bmod (x^n + c)$.

To describe convolutional codes of rate $R = 1/v$, we resurrect a notation used by Massey [6]. The sequence $i_0, i_1, i_2, \cdots$ of information bits is described by its D-transform

$$I(D) = i_0 + i_1 D + i_2 D^2 + \cdots$$

and the convolutional code is defined by the polynomial

$$G(D) = G_1(D^v) + DG_2(D^v) + \cdots + D^{v-1}G_v(D^v). \quad (15)$$

(The component polynomials $G_j(D)$ are now commonly called the "code-generating polynomials" of the convolutional code [5].) If M is the maximum of the degrees of the code-generating polynomials, then M is called the *memory* of the code and $n_A = (M + 1)v$ is called the *constraint length*. The encoded sequence is the sequence $t_0, t_1, t_2, \cdots$ whose D-transform is given by

$$T(D) = I(D^v)G(D), \quad (16)$$

which of course is a polynomial whenever $I(D)$ is a polynomial.

The *free distance* d_{free} of the convolutional code is the minimum of $W[T(D)]$ taken over all $I(D) \neq 0$. The code is said to be *catastrophic*, or to exhibit catastrophic error propagation, if a nonpolynomial $I(D)$ can result in a polynomial $T(D)$ [7], [8]. The well-known necessary and sufficient condition [7] for an $R = 1/v$ code to be non-catastrophic is that

$$\gcd\{G_1(D), G_2(D), \cdots G_v(D)\} = 1, \quad (17)$$

where gcd denotes greatest common divisor and where we have assumed without loss of essential generality that at least one of the code-generating polynomials has a nonzero constant term. It has been well established that d_{free} is the fundamental determinant of error probability for maximum-likelihood (Viterbi) decoding of convolutional codes [9] and

for "almost" maximum-likelihood decoding such as sequential decoding [10].

We now proceed with the use of Theorem 1.3 as a tool for constructing binary convolutional codes with large d_{free} from binary cyclic codes. In what follows, we shall always write $g(x)$ for the generator polynomial of a cyclic code, d_g for the minimum distance of such a code, $h(x) = (x^n + 1)/g(x)$ for the dual polynomial, d_h for the minimum distance of the dual code, and n for the length of both codes.

Theorem 3: If $g(x)$ generates a cyclic code over $GF(2^r)$ of odd length n, then for any positive integer m the rate $R = 1/\upsilon$ 2^r-ary convolutional code with $\upsilon = 2m$ defined by $G(D) = g(D)$ is noncatastrophic and has $d_{\text{free}} \geq \min\{d_g, 2d_h\}$.

Proof: Since n is odd, $g(x)$ has no repeated roots. But

$$G(D) = g(D) = \sum_{j=1}^{\upsilon} D^{j-1} G_j(D^{2m}) = \sum_{j=1}^{\upsilon} D^{j-1} \hat{G}_j(D^m)^2 \quad (18)$$

where we use $\hat{G}_j(D)$ to denote the polynomial obtained from $G_j(D)$ by replacing each coefficient by its square root, the square root existing and being unique for every element in $GF(2^r)$. From (18), we see that any irreducible polynomial that divided each code-generating polynomial would result in an irreducible factor of $g(x)$ with multiplicity of at least 2. We conclude then that gcd $\{G_1(D), G_2(D), \cdots G_\upsilon(D)\} = 1$ so that the convolutional code is noncatastrophic.

For any polynomial $I(D) \neq 0$, we may write

$$T(D) = I(D^{2m})G(D) = \hat{I}(D^m)^2 g(D) \quad (19)$$

where again the coefficients in $\hat{I}(D)$ are the square roots of those in $I(D)$. It then follows from (19) that

$$T(D) = P(D)g(D)^{2i+1}h(D)^{2j} \quad (20)$$

where $i \geq 0$, $j \geq 0$, and $P(D)$ is a nonzero polynomial divisible by neither $g(D)$ nor $h(D)$.

Suppose first that $i \geq j$; then from (20)

$$T(D) = P(D)g(D)^{2(i-j)+1}(D^n + 1)^{2j},$$

which by Theorem 1.3 implies

$$W[T(D)] \geq W[(D+1)^{2j}] \cdot W[P(D)g(D)^{2(i-j)+1} \bmod (D^n + 1)].$$

The first factor on the right is at least 1; further $P(D)g(D)^{2(i-j)+1} \bmod (D^n + 1)$ is a nonzero codeword in the cyclic code generated by $g(X)$ and thus has Hamming weight at least d_g so that

$$W[T(D)] \geq d_g. \quad (21)$$

Conversely, suppose $i < j$. From (20), we then have

$$T(D) = P(D)h(D)^{2(j-i)-1}(D^n + 1)^{2i+1},$$

which by Theorem 1.3 implies

$$W[T(D)] \geq W[(D+1)^{2i+1}] \cdot W[P(D)h(D)^{2(j-i)-1} \bmod (D^n + 1)].$$

The first factor on the right is at least two; the argument of the second factor is a nonzero codeword in the cyclic code generated by $h(x)$ and thus has Hamming weight at least d_h so that

$$W[T(D)] \geq 2d_h. \quad (22)$$

The theorem now follows from (21) and (22).

It should be noted that the lower bound on d_{free} provided by Theorem 3 is independent of m and hence of the rate R of the convolutional code derived from the cyclic code. Hence the bound will be tightest for $m = 1$, i.e., $R = \frac{1}{2}$, since the actual d_{free} can only increase as m increases. The best convolutional codes are obtained by selecting a cyclic code such that $d_g \simeq 2d_h$. In Table I we list several binary ($r = 1$) convolutional codes obtained from Theorem 3 for both $R = \frac{1}{2}$ and $R = \frac{1}{4}$ and indicate the specific cyclic code used in the construction.

The following theorem indicates a somewhat less obvious way to construct convolutional codes from cyclic codes.

Theorem 4: If $g(x)$ generates a cyclic code over $GF(2^r)$ of odd length n, then for any positive integer m the rate $R = 1/\upsilon$ 2^r-ary convolutional code with $\upsilon = 4m$ defined by $G(D) = g(D)^2 + h(D)^2$ is noncatastrophic and has $d_{\text{free}} \geq \min\{d_g + d_h, 3d_g, 3d_h\}$.

Proof: Again we note that since n is odd $g(x)$ has no repeated roots. Moreover, by our choice of $G(D)$, we have from (15)

$$g(D)^2 = \sum_{j=1}^{2m} D^{2(j-1)} G_{2j-1}(D^{4m}) = \sum_{j=1}^{m} D^{2(j-1)} \hat{\hat{G}}_{2j-1}(D^m)^4$$

where we use $\hat{\hat{G}}_j(D)$ to denote the polynomial obtained from $G_j(D)$ by replacing each coefficient by its fourth root, the fourth root existing and being unique for every element of $GF(2^r)$. Hence a common divisor of $G_1(D), G_3(D), \cdots, G_{\upsilon-1}(D)$ would imply that $g(x)$ has some roots of multiplicity exceeding 1. We conclude that gcd $\{G_1(D), G_3(D), \cdots, G_{\upsilon-1}(D)\} = 1$ and hence, *a fortiori*, gcd $\{G_1(D), G_2(D), \cdots, G_\upsilon(D)\} = 1$ so that the convolutional code is noncatastrophic.

For any polynomial $I(D) \neq 0$, we may write

$$T(D) = I(D^{4m})G(D) = \hat{\hat{I}}(D^m)^4[g(D^2) + Dh(D^2)]$$

where the coefficients in $\hat{\hat{I}}(D)$ are the fourth roots of those in $I(D)$. We may then further write

$$T(D) = P(D)g(D)^{4i}h(D)^{4j}[g(D)^2 + Dh(D)^2]$$

where $P(D)$ is a nonzero polynomial divisible by neither $g(D)$ nor $h(D)$, from which it follows that

$$W[T(D)] = W[P(D)g(D)^{4i+2}h(D)^{4j}] + W[P(D)g(D)^{4i}h(D)^{4j+2}]. \quad (23)$$

Suppose first that $i > j \geq 0$. Applying Theorem 1.3 to the second term on the right in (23), we obtain

$$W[P(D)g(D)^{4(i-j)-2}(D^n + 1)^{4j+2}] \geq W[(D+1)^{4j+2}] \cdot W[P(D)g(D)^{4(i-j)-2} \bmod (D^n + 1)].$$

TABLE I
SOME BINARY CONVOLUTIONAL CODES OBTAINED FROM THE CONSTRUCTION GIVEN IN THEOREM 3.

Cyclic Code Employed	R	n_A	d_{free}
(7, 4) QR code, $d_g = 3$, $d_h = 4$	$\frac{1}{2}$	4	≥ 3
	$\frac{1}{4}$	4	≥ 3
(7, 3) QR code, $d_g = 4$, $d_h = 3$	$\frac{1}{2}$	6	≥ 4
	$\frac{1}{4}$	8	≥ 4
(15, 8) BCH code, $d_g = 4$, $d_h = 5$	$\frac{1}{2}$	8	≥ 4
	$\frac{1}{4}$	8	≥ 4
(17, 8) QR code, $d_g = 6$, $d_h = 5$	$\frac{1}{2}$	10	≥ 6
	$\frac{1}{4}$	12	≥ 6
(15, 5) BCH code, $d_g = 7$, $d_h = 4$	$\frac{1}{2}$	12	≥ 7
	$\frac{1}{4}$	12	≥ 7
(23, 12) QR code, $d_g = 7$, $d_h = 8$	$\frac{1}{2}$	12	≥ 7
	$\frac{1}{4}$	12	≥ 7
(23, 11) QR code, $d_g = 8$, $d_h = 7$	$\frac{1}{2}$	14	≥ 8
	$\frac{1}{4}$	16	≥ 8
(31, 11) BCH code, $d_g = 11$, $d_h = 6$	$\frac{1}{2}$	22	≥ 11
	$\frac{1}{4}$	24	≥ 11
(47, 24) QR code, $d_g = 11$, $d_h = 12$	$\frac{1}{2}$	24	≥ 11
	$\frac{1}{4}$	24	≥ 11
(47, 23) QR code, $d_g = 12$, $d_h = 11$	$\frac{1}{2}$	26	≥ 12
	$\frac{1}{4}$	28	≥ 12
(63, 30) BCH code, $d_g = 13$, $d_h \geq 8$	$\frac{1}{2}$	34	≥ 13
	$\frac{1}{4}$	36	≥ 13
(63, 24) BCH code, $d_g = 15$, $d_h \geq 8$	$\frac{1}{2}$	40	≥ 15
	$\frac{1}{4}$	40	≥ 15
(79, 40) QR code, $d_g = 15$, $d_h = 16$	$\frac{1}{2}$	40	≥ 15
	$\frac{1}{4}$	40	≥ 15
(79, 39) QR code, $d_g = 16$, $d_h = 15$	$\frac{1}{2}$	42	≥ 16
	$\frac{1}{4}$	44	≥ 16
(89, 44) QR code, $d_g = 18$, $d_h = 17$	$\frac{1}{2}$	46	≥ 18
	$\frac{1}{4}$	48	≥ 18
(103, 52) QR code, $d_g = 19$, $d_h = 20$	$\frac{1}{2}$	52	≥ 19
	$\frac{1}{4}$	52	≥ 19
(103, 51) QR code, $d_g = 20$, $d_h = 19$	$\frac{1}{2}$	54	≥ 20
	$\frac{1}{4}$	56	≥ 20

The first factor on the right is at least 2 and the second at least d_g, since the argument is a nonzero codeword in the cyclic code generated by $g(x)$. Hence the second term on the right in (23) is at least $2d_g$. A similar argument shows that the first term is at least d_g so that

$$W[T(D)] \geq 3d_g. \tag{24}$$

By an entirely similar argument when $j > i \geq 0$, we have

$$W[T(D)] \geq 3d_h. \tag{25}$$

Finally, suppose that $i = j \geq O$. Applying Theorem 1.3 to the first term on the right in (23), we obtain

$$\begin{aligned} W[P(D)g(D)^2(D^n + 1)^{4i}] &\geq W[(D + 1)^{4i}] \cdot W[P(D)g(D)^2 \bmod (D^n + 1)] \\ &\geq d_g. \end{aligned}$$

A similar argument shows the second term on the right in (23) to be bounded below by d_h so that we have

$$W[T(D)] \geq d_g + d_h. \tag{26}$$

The theorem now follows from (24), (25), and (26).

Again we note that the lower bound on d_{free} provided by Theorem 4 is independent of m and hence of the rate R of the convolutional code derived from the cyclic code. Hence the bound will be tightest for $m = 1$, i.e., for $R = \frac{1}{4}$. The best convolutional codes are obtained by selecting a cyclic code such that $d_g \simeq d_h$. In Table II we list several binary (i.e., $r = 1$) convolutional codes obtained from Theorem 4 for $R = \frac{1}{4}$ and indicate the specific cyclic code used in the construction. Comparison of Tables I and II show that the $R = \frac{1}{4}$ codes obtained from Theorem 4 are often superior to the $R = \frac{1}{4}$ codes obtained from Theorem 3 (or at least the lower bound on d_{free} is larger.)

To obtain an indication of the quality of the convolutional codes obtained by the above constructions, the $n_A = 40$ and $R = \frac{1}{2}$ code of Table I was compared with the $n_A = 40$ and $R = \frac{1}{2}$ Bahl–Jelinek "complementary" code [11] and with the $n_A = 40$ and $R = \frac{1}{2}$ Massey–Costello "quick-look-in" code [10] in Fano-alogirthm [12] sequential decoding for simulated binary symmetric channels and an additive white Gaussian noise channel. In an extensive simulation when the computational cutoff rate R_{comp} of the channel was near the code rate $\frac{1}{2}$, it was found that the Table I code was slightly inferior in undetected error probability to the Bahl–Jelinek code but was significantly superior to the Massey–Costello code. In erasure probability, the Table I code was inferior to the Massey–Costello code but superior to the Bahl-Jelinek code. It seems reasonable then that the codes obtained from Theorems 3 and 4 will be competitive with the best known codes of other constructions.

It should be evident that the generality of Theorem 1.3 admits the construction of many new classes of convolutional codes by mixing $g(x)$ and $h(x)$ from several codes, etc. We leave such extensions to the reader. It should also be remarked that the binary ($r = 1$) $R = \frac{1}{2}$ special case of Theorem 1.3 was independently found by Rudolph and Miczo [13] with a very different argument.

E. Construction of Binary Convolutional Codes from Reed–Solomon Codes

By choosing $g(x)$ in Theorem 3 (or Theorem 4) to be the generator polynomial of a Reed–Solomon (RS) 2^r-ary code [2, p. 310], we can construct some surprisingly good binary convolutional codes for very long constraint lengths. To obtain binary codes, each digit of the RS code is represented as a binary r-tuple so that the $R = \frac{1}{2}$ 2^r-ary convolutional code with one information digit and two encoded digits per subblock becomes an $R = \frac{1}{2}$ binary code with r information bits and $2r$ encoded bits per subblock. The constraint length n_{Ab} of the binary code is r times the constraint length n_A of the 2^r-ary code.

For an $(n = 2^r - 1, k)$ RS code, $d_g = n - k + 1$ and $d_h = k + 1$. Thus the best bound on d_{free} in Theorem 3 results from the choice $k = \lfloor n/3 \rfloor$ where $\lfloor \cdot \rfloor$ denotes the integer part of the enclosed expression. For this choice one obtains

$$d_{\text{free}} \geq \lfloor (2n + 4)/3 \rfloor$$

and

$$n_{Ab} = \begin{cases} r(n - k + 1), & n - k \text{ odd}; \\ r(n - k + 2), & n - k \text{ even}, \end{cases}$$

so that

$$d_{\text{free}}/n_{Ab} \gtrsim 1/r. \tag{27}$$

TABLE II
SOME BINARY CONVOLUTIONAL CODES OBTAINED FROM THE CONSTRUCTION GIVEN IN THEOREM 4.

Cyclic Code Employed	R	n_A	d_{free}
(7, 3) QR code, $d_g = 4$, $d_h = 3$	$\frac{1}{4}$	12	≥ 7
(17, 8) QR code, $d_g = 6$, $d_h = 5$	$\frac{1}{4}$	20	≥ 11
(23, 11) QR code, $d_g = 8$, $d_h = 7$	$\frac{1}{4}$	28	≥ 15
(41, 20) QR code, $d_g = 10$, $d_h = 9$	$\frac{1}{4}$	44	≥ 19
(47, 23) QR code, $d_g = 12$, $d_h = 11$	$\frac{1}{4}$	52	≥ 23
(63, 30) BCH code, $d_g = 13$, $d_h \geq 8$	$\frac{1}{4}$	68	≥ 21
(79, 39) QR code, $d_g = 16$, $d_h = 15$	$\frac{1}{4}$	84	≥ 31
(89, 44) QR code, $d_g = 18$, $d_h = 17$	$\frac{1}{4}$	92	≥ 35
(103, 51) QR code, $d_g = 20$, $d_h = 19$	$\frac{1}{4}$	108	≥ 39

Inequality (27), for $r = 10$, shows that the free distance of these binary codes is still at least 10 percent of the constraint length for $n_{Ab} \simeq 7000$.

Even better codes can be obtained by adding a single parity digit to the r-tuple used to represent the digits of $GF(2^r)$. In this case, the binary code still has only r information bits per subblock but the subblock length is increased to $2(r + 1)$ and hence the rate of the binary code is reduced to

$$R = \frac{1}{2}\,\frac{r}{r+1}, \tag{28}$$

which approaches $\frac{1}{2}$ for large r. Since there are at least two nonzero bits in each nonzero digit of $GF(2^r)$ in this new representation, one has for the binary code

$$d_{\text{free}} \geq 2\lfloor (2n + 4)/3 \rfloor$$

and also

$$n_{Ab} = \begin{cases} (r+1)(n-k+1), & n-k \text{ odd}; \\ (r+1)(n-k+2), & n-k \text{ even}, \end{cases}$$

so that

$$d_{\text{free}}/n_{Ab} \gtrsim 2/(r+1). \tag{29}$$

Inequality (29), for $r = 19$ and hence R quite near $\frac{1}{2}$, shows that the free distance of these binary codes is still at least 10 percent of the constraint length for $n_{Ab} \simeq 6\ 700\ 000$.

The strongest known lower bound on d_{free} for binary convolutional codes is that of Neumann [14] but there is an improved lower bound due to Costello [15] for "time-varying convolutional codes." For $R = \frac{1}{2}$, these lower bounds on d_{free}/n_A become 0.22 and 0.40, respectively for large n_A. The lower bound on d_{free}/n_{Ab} for the second class of codes in this section remains above these values for $n_{Ab} \leq 900$ and $n_{Ab} \leq 100$, respectively.

III. THE NONBINARY CASE

Hereafter, p shall denote a prime greater than 2, c a nonzero element of $GF(p^r)$, n the length of a p^r-ary block code, k the number of information digits in said code, and d the minimum distance of said code.

A. A New Class of p^r-ary Repeated-Root Constacyclic Codes With an Algebraic Decoding Algorithm

Following Berlekamp's terminology [2, p. 303], we shall say that a polynomial $g(x)$ over $GF(p^r)$ of degree $n - k$, which divides $x^n - c^p$, generates an (n, k) *constacyclic* code whose codewords are all the multiples of $g(x)$ having degree less than n. The code is cyclic if and only if $c = 1$ and is *negacyclic* [2, p. 211] if and only if $c = -1$. The following theorem gives a new class of repeated-root constacyclic codes and in its proof we develop an algebraic decoding algorithm for these codes. The cyclic codes in this class have been given earlier by Assmus and Mattson [16] and by Berman [17].

Theorem 5: The polynomial $g(x) = (x - c)^{p-k}$ for $1 \leq k < p$ generates a p^r-ary $(n = p, k)$ constacyclic code with $d = n - k + 1$ (i.e., a *maximum distance separable* code [2, p. 309].)

Proof: We note first that $(x - c)^p = x^p - c^p$ so that $g(x)$ divides $x^p - c^p$ and hence generates a constacyclic code of length $n = p$. Moreover, by Lemma 1, $W[g(x)] = p - k + 1 = n - k + 1$ so that $d \leq n - k + 1$.

Let $f(x)$ be a codeword in this constacyclic code and let $e(x) = e_0 + e_1 x + \cdots + e_{p-1}x^{p-1}$ be the channel error pattern. The same analysis as led to (11) above now shows that after $f(x) + e(x)$ is read into the circuit of Fig. 2 with higher degree terms leading then the resultant syndrome is given by

$$s(x) = e(x + c) \bmod (x^{p-k}) \tag{30a}$$

or equivalently

$$s(x) = \sum_{j=0}^{p-1} e_j (x + c)^j \bmod (x^{p-k}). \tag{30b}$$

From (30b), we see that the syndrome $s(x) = s_0 + s_1 x + \cdots + s_{p-k-1}x^{p-k-1}$ can be expressed as

$$s_i = \sum_{j=0}^{p-1} \binom{j}{i} c^{j-i} e_j, \qquad 0 \leq i < p - k \tag{31a}$$

or equivalently

$$s_i = (i!)^{-1}\, \partial^i e(c), \qquad 0 \leq i < p - k. \tag{31b}$$

where we write $\partial^i e(c)$ for the ith formal derivative of $e(x)$ evaluated at $x = c$.

We now define modified syndrome digits $S_0, S_1, \cdots, S_{p-k-1}$ as the following linear combination of the s_i

$$S_i = \sum_{j=1}^{i} \left\{ {i \atop j} \right\} (j!) c^i s_i, \qquad 1 \leq i < p - k \tag{32}$$

and

$$S_0 = s_0 \tag{33}$$

where $\left\{ {i \atop j} \right\}$ denotes the Stirling number of the second kind [19]. From (31a) and (32) we obtain

$$S_i = \sum_{j=1}^{i} \sum_{m=0}^{p-1} c^m (j!) \binom{m}{j} e_j, \qquad 1 \leq i < p - k$$

so that we may write

$$S_i = \sum_{j=0}^{p-1} e_j c^j j^i, \qquad 0 \leq i < p - k \tag{34}$$

with the understanding that $0^0 = 1$.

Suppose that the error pattern $e(x)$ has weight t so that

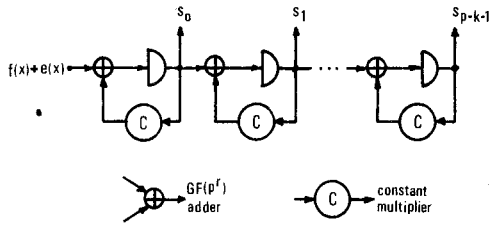

Fig. 2. Syndrome-forming circuit for the p^r-ary repeated-root constacyclic codes of Theorem 5.

it may be written

$$e(x) = \sum_{j=1}^{t} Y_j(x/c)^{i_j}$$

where $Y_j \neq O$ is the "modified" *value* of the jth error (and is related to the true error value e_{i_j} as $e_{i_j} = Y_j c^{-i_j}$) and where we define $X_j = i_j$ as the *location* of the jth error. Equation (34) may then be rewritten as

$$S_i = \sum_{j=1}^{t} Y_j X_j^i, \qquad 0 \leq i < p - k. \tag{35}$$

But (35) is just the usual syndrome relations for a BCH code with "design distance" [2, p. 274] $p - k + 1$, which implies then that $d \geq p - k + 1$.

We conclude then that $d = p - k + 1$, which proves the theorem.

Moreover, we conclude from (35) that the error values and locations for the constacyclic codes of Theorem 5 may be determined by any BCH decoding procedure, such as Berlekamp's iterative algorithm [2, pp. 219–221]. Particularly when p is a Mersenne prime, i.e., when $p = 2^m - 1$ for some integer m, so that $GF(p)$ operations are just "one's complement" arithmetic, the decoding procedure for the p-ary codes would be easy to implement, especially when $c = 1$ (the cyclic codes) or $c = -1$ (the negacyclic codes) so that the syndrome-former in Fig. 2 is very simple. These codes might find practical application in concatenated coding schemes [19].

We remark further that the parameters n, k, and d of the constacyclic codes of Theorem 5 coincide with those of the "extended" [2, p. 234] p-ary Reed–Solomon codes and for $c = 1$ are just permutations of these codes. The interesting difference is that the repeated-root codes permit the use of the integers $0,1,\cdots,p-1$ as the error locations whereas in the RS codes the error locators are taken as 0 and the powers $\alpha^0,\alpha,\cdots,\alpha^{p-2}$ of a primitive element α of $GF(p)$. Use of the additive group of $GF(p)$, rather than the multiplicative group, as the error locations results both in the natural inclusion of the "0" position and also in a reduction in the number of multiplications required for decoding by the iterative algorithm.

B. The Weight-Retaining Property of $(x - c)^i$ over $GF(p^r)$

Again we remark that the polynomials $(x - c)^i$, $i = 0,1,2,\cdots$, form a basis for the vector space of all polynomials over $GF(p^r)$. We now propose to show that the weight-retaining property given in Theorem 1.1 for $p = 2$ holds in general. We find it more convenient to prove first the p-ary analog of Theorem 1.2 and thereafter to deduce the p-ary analog of Theorem 1.1.

Theorem 6.2: For any polynomial $Q(x)$ over $GF(p^r)$, any nonzero c in $GF(p^r)$, and any nonnegative integer N,

$$W[Q(x)(x - c)^N] \geq W[(x - c)^N] \cdot W[Q(c)]. \tag{36}$$

Proof: In what follows we shall make frequent use of the fact that for any i and any polynomial $P(x), W[P(x)] \geq W[P(x) \bmod (x^i - c)]$.

We first show that (36) holds for $N < p$. If $Q(c) = 0$, then (36) holds trivially. If $Q(c) \neq 0$, then $Q(x)$ is not divisible by $(x - c)$ so that $Q(x)(x - c)^N \bmod (x^p - c^p)$ is a nonzero codeword in the constacyclic code generated by $g(x) = (x - c)^N$ and hence, by Theorem 5, has Hamming weight at least $N + 1$. Thus

$$\begin{aligned} W[Q(x)(x - c)^N] &\geq W[Q(x)(x - c)^N \bmod (x^p - c^p)] \\ &\geq N + 1 = W[(x - c)^N] \end{aligned}$$

where the last inequality follows from Lemma 1. Hence (36) holds for $N < p$.

We now suppose that (36) holds for $N < Kp^i, 1 \leq K < p$, and proceed by induction on K, which also includes induction on i since $(K + 1)p^i = p^{i+1}$ when $K = p - 1$. We have already shown that (36) holds for $i = 0$ so that a basis has been established for the induction. It remains to show that (36) holds for $N < (K + 1)p^i$ or, equivalently, for $N = Kp^i + L$ for all integers L such that $0 \leq L < p^i$.

We begin by noting that

$$\begin{aligned} W[Q(x)(x - c)^N] &= W[Q(x)(x - c)^L(x - c)^{Kp^i}] \\ &= W[Q(x)(x - c)^L(x^{p^i} - c^{p^i})^K]. \end{aligned} \tag{37}$$

Now writing

$$P(x) = Q(x)(x - c)^L = \sum_{j=0}^{p^i-1} x^j P_j(x^{p^i}) \tag{38}$$

we have then from (37)

$$W[Q(x)(x - c)^N] = \sum_{j=0}^{p^i-1} W[P_j(x)(x - c^{p^i})^K] \tag{39}$$

where we have merely replaced x^{p^i} by x in (38). Since $K < p$, (36) holds for each term on the right in (39) and, together with Lemma 1, gives

$$W[Q(x)(x - c)^N] \geq (K + 1) \sum_{j=0}^{p^i-1} W[P_j(c^{p^i})],$$

which, upon invoking (38), may be written as

$$\begin{aligned} &W[Q(x)(x - c)^N] \\ &\quad \geq (K + 1)W[Q(x)(x - c)^L \bmod (x^{p^i} - c^{p^i})]. \end{aligned} \tag{40}$$

Now also

$$\begin{aligned} &Q(x)(x - c)^L \bmod (x^{p^i} - c^{p^i}) \\ &\qquad = Q(x)(x - c)^L \bmod (x - c)^{p^i} \\ &\qquad = [Q(x) \bmod (x - c)^{p^i - L}](x - c)^L. \end{aligned} \tag{41}$$

But $L < p^i$ so that (36) may be applied to the right-hand side of (41) to give

$$W[Q(x)(x-c)^L \bmod (x^{p^i}-c^{p^i})] \geq W[(x-c)^L]\cdot W[Q(c)] \quad (42)$$

where we have recognized that $Q(x) \bmod (x-c)^L$ evaluated at $x = c$ is just $Q(c)$. From (40), (41), and (42), we obtain

$$W[Q(x)(x-c)^N] \geq (K+1)W[(x-c)^L]\cdot W[Q(c)],$$

which, with the aid of Lemma 1 and the facts that $L < Kp^i$ and $N = Kp^i + L$, may finally be written as

$$W[Q(x)(x-c)^N] \geq W[(x-c)^N]\cdot W[Q(c)],$$

which is (36), and the theorem is proved.

We now have as a trivial consequence of Theorem 6.2:

Theorem 6.1: Let I be any nonempty finite set of nonnegative integers with least integer $i_{\min}$ and let

$$P(x) = \sum_{i\in I} b_i(x-c)^i$$

where c and each b_i are nonzero elements of $GF(p^r)$. Then

$$W[P(x)] \geq W[(x-c)^{i_{\min}}]. \quad (43)$$

This theorem follows from Theorem 6.2 by noting that

$$P(x) = Q(x)(x-c)^{i_{\min}} \text{ where } Q(c) = b_{i_{\min}} \neq 0.$$

Theorem 6.1 is the desired p-ary analog of Theorem 1.1. Although we shall make no further use of it, we now state for completeness the p-ary analog of Theorem 1.3, which follows from Theorem 6.2 precisely as Theorem 1.3 followed from Theorem 1.1.

Theorem 6.3: For any polynomial $P(x)$ over $GF(p^r)$, any nonzero element c of $GF(p^r)$, and any nonnegative integers n and N,

$$W[P(x)(x^n-c)^N] \geq W[(x-c)^N]\cdot W[P(x) \bmod (x^n-c)].$$

C. A New Class of p-ary "Reed–Muller" Codes

Proceeding analogously to Section II-B, let m be any positive integer and consider the matrix G with $n = p^m$ columns, whose rows are the sequences of coefficients of $(x-c)^i$ for all $i < n$ such that $W[(x-c)^i] \geq d$, where d is some integer chosen such that equality holds for at least one such i. Given n and d, the k corresponding integers i can be found with the aid of Lemma 1. For simplicity, one would usually take $c = 1$ or $c = -1$, but this is not necessary. It follows immediately from Theorem 6.1 that G is the generator matrix of an $(n = p^m, k)$ p-ary code with minimum distance d. We call these codes "p-ary Reed–Muller codes" because of their similarity to the binary Reed–Muller codes as formulated in Section II-B. A short list of these codes is given in Table III.

TABLE III
A SHORT TABLE OF THE NEW CLASS OF p-ARY REED-MULLER CODES GIVEN IN SECTION III-C.

p	n	k	d	p	n	k	d
3	9	8	2	5	5	4	2
	9	6	3		5	3	3
	9	4	4		5	2	4
	9	3	6		5	1	5
	9	1	9		25	24	2
	27	26	2		25	22	3
	27	23	3		25	20	4
	27	20	4		25	17	5
	27	17	6		25	15	6
	27	11	8		25	13	8
	27	10	9		25	11	9
	27	7	12		25	10	10
	27	4	18		25	8	12
	27	1	27		25	6	15
					25	4	16
					25	3	20
					25	1	25

The codes in Table III have in most instances the same parameters n, k, and d as do the extended p-ary Reed–Muller codes as given by Kasami *et al.* [20] (which we hereafter call KLP codes) whenever the more sparse KLP codes exist; i.e., the codes of Table III generally have the same k but n and d both one larger than the corresponding KLP code. However, the 5-ary (25, 15) code in Table III has $d = 6$, whereas d is only 4 for the 5-ary (24, 15) KLP code, so that the "p-ary Reed–Muller codes" as given here are not merely permutations of the (more sparse) KLP codes.

We also remark that one can obtain repeated-root constacyclic subcodes of the p-ary Reed–Muller codes given here that are analogous to the cyclic codes of Section II-C. The generator polynomial of the constacyclic code is chosen as $g(x) = (x-c)^{n-k}$, where $k = p^{m-u+1} - 1$. These constacyclic subcodes have the same minimum distance $d = 2p^{u-1}$ as their parent p-ary Reed–Muller codes. The constacyclic code is cyclic if and only if $c = 1$ and is negacyclic if and only if $c = -1$.

IV. Concluding Remarks

In the foregoing, it has been shown that the weight-retaining properties of the polynomials $(x-c)^i$ admit of exploitation in the construction of both block codes and convolutional codes. This generality is somewhat surprising, and it seems safe to say that not all the consequences of the weight-retaining property have been uncovered in this paper.

We wish to remark that the discovery by the second author of the binary $R = \frac{1}{2}$ codes of Theorem 3 was the initial impetus for the research reported here. The binary convolutional codes of Theorem 3 for $m = 2^i$ and of Theorem 4 for $m = 1$ were earlier described orally by the first two authors [21]. In the original manuscript of this paper, the results were obtained only for prime fields $GF(p)$ rather than $GF(p^r)$ as appears here. The most significant consequence of this generalization was the subsequent discovery of the long constraint length convolutional codes in Section II-E by Justesen.

ACKNOWLEDGMENT

The authors are grateful to Dr. G. David Forney, Jr., of the Codex Corporation for his keen interest in and encouragement of this research and for his suggestion of the simple proof of Lemma 1. (An alternative proof of Lemma 1 may be found in Berman [17].)

REFERENCES

[1] F. P. Preparata, "State-logic relations for autonomous sequential networks," *IEEE Trans. Electron. Comput.*, vol. EC-13, pp. 542–548, Oct. 1964.
[2] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.
[3] I. S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IRE Trans. Inform. Theory*, vol. PGIT-4, pp. 38–49, Sept. 1954.
[4] J. E. Meggitt, "Error correcting codes and their implementation for data transmission systems," *IRE Trans. Inform. Theory*, vol. IT-7, pp. 234–244, Oct. 1961.
[5] J. L. Massey, *Threshold Decoding*. Cambridge, Mass.: M.I.T. Press, 1963.
[6] ——, "Majority decoding of convolutional codes," Res. Lab. Electron. Mass. Inst. Technol., Cambridge, Quart. Prog. Rep. 64, pp. 183–188, Jan. 15, 1962.
[7] J. L. Massey and M. K. Sain, "Inverses of linear sequential circuits," *IEEE Trans. Comput.*, vol. C-17, pp. 330–337, Apr. 1968.
[8] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.
[9] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 751–772, Oct. 1971.
[10] J. L. Massey and D. J. Costello, Jr., "Nonsystematic convolutional codes for sequential decoding in space applications," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 806–813, Oct. 1971.
[11] L. R. Bahl and F. Jelinek, "Rate $\frac{1}{2}$ convolutional codes with complementary generators," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 718–727, Nov. 1971.
[12] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. IT-9, pp. 64–74, Apr. 1963.
[13] L. D. Rudolph and A. Miczo, "Some results on the distance properties of convolutional codes," Syracuse Univ., Syracuse, N.Y., Final Rep. NSF Grant GK-4737, Oct. 1970.
[14] B. Neumann, "Distance properties of convolutional codes," S.M. thesis, Dep. Elec. Eng., Mass. Inst. Technol., Cambridge, Aug. 1968.
[15] D. J. Costello, Jr., "Construction of convolutional codes for sequential decoding," Dep. Elec. Eng., Univ. Notre Dame, Notre Dame, Ind., Tech. Rep. EE-692, Aug. 1969.
[16] E. F. Assmus, Jr., and H. F. Mattson, Jr., "New 5-designs," *J. Combinatorial Theory*, vol. 6, pp. 122–151, 1969.
[17] S. D. Berman, "On the theory of group codes," *Kibernetika*, vol. 3, no. 1, pp. 31–39, 1967.
[18] J. Riordan, *An Introduction to Combinatorial Analysis*. New York: Wiley, p. 33, 1958.
[19] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, Mass.: M.I.T. Press, 1966.
[20] T. Kasami, S. Lin, and W. W. Peterson, "New generalizations of the Reed–Muller codes," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 189–198, Jan. 1968.
[21] D. J. Costello, Jr., and J. L. Massey, "Constructing good convolutional codes from cyclic block codes," presented at IEEE Int. Symp. Information Theory, Asilomar, Calif., Jan. 1972.

Correspondence

A General Approach to Linear Mean-Square Estimation Problems

STAMATIS CAMBANIS

Abstract—**An explicit and easily implemented solution is given to the general problem of linear mean-square estimation of a signal or system process based upon noisy observations, under the assumption that the auto- and cross-correlation functions of the signal and the observation processes are known. Also a number of specific estimation problems are briefly discussed.**

I. INTRODUCTION

In this correspondence the linear mean-square estimation of a signal or system process on the basis of noisy observations is considered. A general solution is obtained under the assumption only that the autocorrelation of the observation process and the cross-correlation between the observation and the signal processes are known. No assumptions are made about the continuous, stationary, or Gaussian properties of the signal or of the observation processes, or about the nature of the observation interval. The estimate is expressed in a series, each term of which is an easily implemented linear operation on the realizations of the observation process. The estimate is also approximated arbitrarily closely in the stochastic mean-square sense by an explicitly given linear integral operation on the observation process. The significance of this estimation procedure is that it provides a general solution to linear mean-square estimation problems that can be implemented in a straightforward way.

The basis for the approach taken in this correspondence is some structural properties of second-order processes presented in [1]. These properties are summarized in Section II. In Section III the general linear mean-square estimation problem is formulated and solved. A number of estimation problems for which the estimation procedure presented in Section III is applicable are discussed in Section IV.

It should be remarked that the use of stochastic-process representations in the solution of linear mean-square estimation problems is well known and widely used; see for instance [4] and further references mentioned in the following. In the light of prior work on this subject the contribution of this correspondence is twofold: i) the linear mean-square estimation problem is solved for all signal or system processes and observation processes that are measurable and of second order and for all observation intervals; in contrast, all solutions previously reported in the literature apply to more restrictive classes of processes and observation intervals; and ii) the estimate can be implemented in a straightforward way and a significant freedom

Manuscript received May 4, 1971; revised April 12, 1972. This work was supported by the Air Force Office of Scientific Research under Grant AFOSR-68-1415.
The author is with the Department of Statistics, University of North Carolina at Chapel Hill, Chapel Hill, N.C. 27514.